

#260  
12-26-01

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

03/13/01  
09/804302  
U.S. PTO  
JC872

In re PATENT APPLICATION of  
Inventor(s): Toru OTSUKA

Appln. No.: To Be Assigned  
Series ↑      |  
Code      | ↑ Serial No.

Group Art Unit: To Be Assigned

Filed: March 13, 2001

Examiner: To Be Assigned

Title: SIMULATOR AND SIMULATION METHOD

Atty. Dkt. P 278088 | T4A0-00A0902-1  
M# | Client Ref

Date: March 13, 2001

**SUBMISSION OF PRIORITY  
DOCUMENT IN ACCORDANCE  
WITH THE REQUIREMENTS OF RULE 55**

Hon. Asst Commissioner of Patents  
Washington, D.C. 20231

Sir:

Please accept the enclosed certified copy(ies) of the respective foreign application(s) listed below for which benefit under 35 U.S.C. 119/365 has been previously claimed in the subject application and if not is hereby claimed.

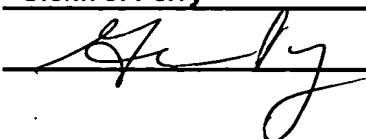
<u>Application No.</u>	<u>Country of Origin</u>	<u>Filed</u>
------------------------	--------------------------	--------------

2000-069232	JAPAN	March 13, 2000
-------------	-------	----------------

Respectfully submitted,

**Pillsbury Winthrop LLP  
Intellectual Property Group**

1100 New York Avenue, NW  
Ninth Floor  
Washington, DC 20005-3918  
Tel: (202) 861-3000  
Atty/Sec: gjp/mjb

By Atty: <u>Glenn J. Perry</u>	Reg. No. <u>28458</u>
Sig: 	Fax: (202) 822-0944 Tel: (202) 861-3070

日本国特許庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

JC872 U.S. PTO  
09/804302  
03/13/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application: 2000年 3月13日

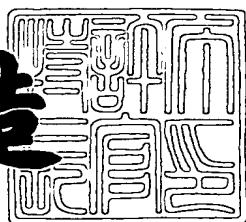
出願番号  
Application Number: 特願2000-069232

出願人  
Applicant(s): 株式会社東芝

2000年 9月 1日

特許庁長官  
Commissioner,  
Patent Office

及川耕造



出証番号 出証特2000-3070731

【書類名】 特許願  
【整理番号】 A000000494  
【提出日】 平成12年 3月13日  
【あて先】 特許庁長官 殿  
【国際特許分類】 G06F 11/00  
【発明の名称】 シミュレータ及びシミュレーション方法  
【請求項の数】 10  
【発明者】  
【住所又は居所】 神奈川県川崎市幸区柳町70番地 株式会社東芝柳町工場内  
【氏名】 大塚 徹  
【特許出願人】  
【識別番号】 000003078  
【氏名又は名称】 株式会社 東芝  
【代理人】  
【識別番号】 100058479  
【弁理士】  
【氏名又は名称】 鈴江 武彦  
【電話番号】 03-3502-3181  
【選任した代理人】  
【識別番号】 100084618  
【弁理士】  
【氏名又は名称】 村松 貞男  
【選任した代理人】  
【識別番号】 100068814  
【弁理士】  
【氏名又は名称】 坪井 淳  
【選任した代理人】  
【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 シミュレータ及びシミュレーション方法

【特許請求の範囲】

【請求項1】

シミュレーション用C P Uと、  
当該シミュレータが接続される制御C P U及び前記シミュレーション用C P U  
の一方から書き込みが可能で、他方から読み出しが可能なメモリと、  
前記制御C P Uから前記メモリに書き込まれた制御情報を、前記シミュレーション用C P Uにより読み出す手段と、  
この制御情報に基づくシミュレーションを実行した結果を前記制御C P Uから  
読み出し可能なように前記メモリに書き込む手段とを有する、  
ことを特徴とするシミュレータ。

【請求項2】

シミュレーション用C P Uと、  
当該シミュレータが接続される制御C P U及び前記シミュレーション用C P U  
の一方から書き込みが可能で、他方から読み出しが可能なメモリと、  
前記シミュレーション用C P Uと前記メモリとを接続する接続手段と、  
前記制御C P Uから前記メモリに書き込まれた制御情報を、前記接続手段を介  
して前記シミュレーション用C P Uにより読み出す手段と、  
この制御情報に基づくシミュレーションを実行した結果を前記制御C P Uから  
読み出し可能なように前記接続手段を介して前記メモリに書き込む手段とを有す  
る、  
ことを特徴とするシミュレータ。

【請求項3】

前記シミュレーション用C P Uは、前記シミュレーションの結果を前記メモリ  
に書き込んだ後、前記制御C P Uに対して割り込みを要求する  
ことを特徴とする請求項1又は請求項2に記載のシミュレータ。

【請求項4】

当該シミュレータが接続される制御C P Uから制御情報が書き込まれる第1の

メモリと、

前記制御C P Uから情報が読み出し可能な第2のメモリと、

前記第1のメモリから制御情報を読み出す手段と、

この読み出された制御情報に基づいた制御結果の情報を発生する手段と、

この発生された制御結果の情報を前記第2のメモリに書き込む手段とを有する

ことを特徴とするシミュレータ。

**【請求項5】**

さらに、前記シミュレーションの結果を前記第2のメモリに書き込んだ後、前記制御C P Uに対して割り込みを送信する手段を有する、

ことを特徴とする請求項4に記載のシミュレータ。

**【請求項6】**

制御C P Uにより制御情報を第1のメモリに書き込み、

シミュレーション用C P Uにより前記第1のメモリに書き込まれた前記制御情報を接続手段を介して読み出し、

前記シミュレーション用C P Uにより前記制御情報に基づくシミュレーションを実行し、

前記シミュレーション用C P Uによりシミュレーションの結果を接続手段を介して第2のメモリに書き込み、

前記制御C P Uにより前記第2のメモリに書き込まれた前記シミュレーションの結果を読み出す、

ことを特徴とするシミュレート方法。

**【請求項7】**

当該シミュレータが接続される制御C P Uからコマンドが書き込まれる第1のメモリと、

前記制御C P Uから情報が読み出し可能な第2のメモリと、

前記第1のメモリからコマンドを読み出す手段と、

この読み出されたコマンドに基づいたレスポンスを発生する手段と、

この発生されたレスポンスを前記第2のメモリに書き込む手段とを有する、

ことを特徴とするシミュレータ。

【請求項8】

当該シミュレータが接続される制御C P Uから情報が読み出し可能なメモリと

前記メモリに定期的にセンサ状態を書き込む手段とを有する、

ことを特徴とするシミュレータ。

【請求項9】

当該シミュレータが接続される制御C P Uからコマンドが書き込まれる第1の  
メモリと、

前記制御C P Uから情報が読み出し可能な第2及び第3のメモリと、

前記第1のメモリからコマンドを読み出す手段と、

この読み出されたコマンドに基づいたレスポンスを発生する手段と、

この発生されたレスポンスを前記第2のメモリに書き込む手段と、

前記第3のメモリに定期的にセンサ状態を書き込む手段とを有する、

ことを特徴とするシミュレータ。

【請求項10】

当該シミュレータが接続される制御C P Uから出力ポートのオン／オフが書き  
込まれる第1のメモリと、

前記制御C P Uからコマンドが書き込まれる第2のメモリと、

前記制御C P Uから情報が読み出し可能な第3及び第4のメモリと、

前記第1のメモリから出力ポートのオン／オフを読み出す手段と、

前記第2のメモリからコマンドを読み出す手段と、

この読み出されたコマンドに基づいたレスポンスを発生する手段と、

この発生されたレスポンスを前記第3のメモリに書き込む手段と、

前記第4のメモリに定期的にセンサ状態を書き込む手段とを有する、

ことを特徴とするシミュレータ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、機構制御プログラムを実機無しでデバッグするために、あたかも実機があるかのように反応するシミュレータ及びシミュレート方法に関する。

#### 【0002】

##### 【従来の技術】

機構制御プログラムのデバッグは、機構とそれを駆動制御するハードウェア（機構+ハードウェアを実機と呼ぶ）を作成し、実際に実機を制御してみることにより行われていた。

#### 【0003】

##### 【発明が解決しようとする課題】

しかし、実機を実際に制御して機構制御プログラムをデバッグするという手法では、実機が完成するまで本格的なデバッグができないという問題があった。また、稀にしか起きない異常状態を故意に発生させることが困難であり、十分な検証ができないという問題もあった。

#### 【0004】

この発明の目的は、実機が完成していないあたかも実機があるかのように制御プログラムのデバッグが行え、更に様々な異常状態を任意に発生させて制御プログラムを検証することが可能なシミュレータ及びシミュレート方法を提供することにある。

#### 【0005】

##### 【課題を解決するための手段】

上記課題を解決し目的を達成するために、この発明のシミュレータ及びシミュレート方法は、以下のように構成されている。

#### 【0006】

(1) この発明のシミュレータは、シミュレーション用CPUと、当該シミュレータが接続される制御CPU及び前記シミュレーション用CPUの一方から書き込みが可能で、他方から読み出しが可能なメモリと、前記制御CPUから前記メモリに書き込まれた制御情報を、前記シミュレーション用CPUにより読み出す手段と、この制御情報に基づくシミュレーションを実行した結果を前記制御CPUから読み出し可能なように前記メモリに書き込む手段とを有する。

**【0007】**

(2) この発明のシミュレート方法は、制御CPUにより制御情報を第1のメモリに書き込み シミュレーション用CPUにより前記第1のメモリに書き込まれた前記制御情報を接続手段を介して読み出し、前記シミュレーション用CPUにより前記制御情報に基づくシミュレーションを実行し、前記シミュレーション用CPUによりシミュレーションの結果を接続手段を介して第2のメモリに書き込み、前記制御CPUにより前記第2のメモリに書き込まれた前記シミュレーションの結果を読み出す。

**【0008】****【発明の実施の形態】**

以下、この発明の実施の形態について図面を参照して説明する。

**【0009】**

図1を参照して実機を用いたデバッグを説明し、これに対比させて、図2を参照してこの発明の実機を用いないデバッグ（シミュレータ）を説明する。図1は実機ハードウェアの概略を示す図であり、図2はシミュレータの概略を示す図である。

**【0010】**

図1に示すように、実機では制御CPU101に機構を制御するASIC102(CPU側)が接続されている。このASIC102とASIC103とがシリアル回線104を介して接続されている。さらに、ASIC103にはモータ105、ソレノイド106、及びセンサ107等が接続されている。制御CPU101は、ASIC102のレジスタに制御情報を書き込むことで、ASIC103に接続されたモータ105、ソレノイド106、及びセンサ107等を制御することができる。

**【0011】**

モータ105、ソレノイド106、及びセンサ107等は、ASIC103を介して伝えられる制御CPU1からの制御情報に従って動作し、機構の動作及び媒体の動作等が発生し、結果としてセンサのオン／オフ状態が変化する。制御CPU101は、ASIC102のレジスタを参照することにより、ASIC10

3に接続された各センサの状態を検知することができる。

#### 【0012】

これに対して、図2に示すように、シミュレータでは、シミュレータボード112上で制御CPU113にシミュレータハードウェア114が接続されている。シミュレータハードウェア114は、制御CPU113から見ると、センサ情報参照レジスタ、出力ポート制御情報書き込みレジスタ、モータ制御コマンド書き込みレジスタ、レスポンス情報参照レジスタ、及び割り込みステータスレジスタ等のレジスタ群114aを備えている。

#### 【0013】

これらレジスタ群114aは、ASIC102のレジスタと同じ構成のレジスタであり、制御CPU113及びCPU115の両CPUから読書可能なメモリに相当する。制御CPU113上で実機用の制御プログラムを実行すると、実際にはシミュレータハードウェア114のレジスタ群114aの中の所定のレジスタを参照し、書き込みながら動作する。

#### 【0014】

制御CPU113がレジスタ群114aの中の所定のレジスタに書き込んだ制御情報は、PCIバス116を介してパソコン111のCPU115上で実行されるシミュレータプログラムに伝わる。PCIバスを経由したアクセスは、実際にはドライバが行っており、DLL関数がCPU115とドライバの接続を行うために利用される。

#### 【0015】

シミュレータプログラムは予め定義された機構動作記述を参照し、制御CPU113からの制御情報に従った動作をシミュレーションし、結果としてのセンサの変化をシミュレータハードウェア114のレジスタ群114aの中の所定のレジスタに書き込む。制御CPU113はシミュレータハードウェア114のレジスタ群114aの中の所定のレジスタを参照することで、実機において機構動作の結果としてセンサ状態の変化を参照しているのと全く同様に、シミュレータプログラムが書き込んだシミュレーション結果をセンサ変化として読むことができる。

**【0016】**

この動作の様子を図3を参照して説明する。図3は、シミュレータハードウェアのレジスタ群に対するデータの読み書きの流れ、及び割り込み処理等を示す図である。

**【0017】****1、出力ポート**

制御CPU113が出力ポートに書き込んだデータは、シミュレータハードウェア114内部のDP-RAMに書き込まれ、シミュレータ用ソフトはGetPortStatus()関数をコールすることにより、データを読み出すことができる。

**【0018】****2、コマンド**

制御CPU113が出力ポートに書き込んだデータは、シミュレータハードウェア114内部のDP-RAMに書き込まれ、シミュレータ用ソフトはGetCommand()関数をコールすることにより、書かれたコマンドを読み出すことができる。

**【0019】****3、センサ**

シミュレータ用ソフトからPutSensorStatus()関数をコールすることにより、センサ状態データをシミュレータハードウェア114内のDP-RAMにセットする。制御CPU113はシミュレータハードウェア114の所定アドレスを読み出すことにより、センサ状態のデータを読み出すことができる。

**【0020】****4、レスポンス**

シミュレータ用ソフトからPutResponse()関数をコールすることにより、レスポンスをシミュレータハードウェア114内のDP-RAMにセットする。制御CPU113はシミュレータハードウェア114の所定アドレスを読み出すことにより、レスポンスデータを得ることができる。

**【0021】****5、割込み**

制御CPU113のプログラムはRSCからの割込みをトリガとして処理する。

従って、シミュレータ用ソフト側で  $128 \mu\text{sec}$  分の処理を完了しセンサ状態の更新及びレスポンスの書き込みを行ったら、IssueInterrupt()関数をコールして制御CPU113側に処理を促す必要がある。

#### 【0022】

##### 6. ステータス

シミュレータハードウェア114内の回路は、制御CPU113側およびシミュレータ用ソフト側のアクセスに従って、ASICのステータスレジスタ相当のレジスタを更新する。実際に更新を行なうのは書き込みバッファフル、書き込みバッファエンプティ、受信データ有りの3ビットである。

#### 【0023】

次に、シミュレータプログラム（CPU115で実行）と制御プログラム（制御CPU113で実行）の実行同期について説明する。

#### 【0024】

実機のハードウェアではセンサスキャンは  $100 \mu\text{sec} \sim 1\text{ msec}$  周期で行われているが、シミュレータプログラムはパソコン上で実行するソフトウェアであり、現状のCPU能力ではシミュレーション実行と結果の反映を実機のセンサスキャン周期と同じ時間で実行することは困難である。また、シミュレーションの所要時間はシミュレータ内部の機構の状態により変化するため、制御CPU113との間で同期を取る必要がある。

#### 【0025】

これを実現するために、既述の割込みを用いる。シミュレータ側はシミュレーションが1回完了し、センサ状態およびレスポンスをシミュレータハードウェア114のレジスタ群14aの中の所定のレジスタに書き込んだ後に、割込み発生要求を行う。

#### 【0026】

制御プログラムの構造は図4に示す割り込み処理を実行する。割り込み処理では、割込み、センサ変化のいずれかをトリガとして機構制御処理を実行する。つまり、割り込みが発生すると、ASIC（シミュレータハードウェア）のレスポンスレジスタを参照し、必要処理を実行する。ASIC（シミュレータハードウ

エア) のセンサレジスタを参照し、センサ変化を検出し、対応処理を実行する。そして、ソフトキュー内のコマンドをASIC(シミュレータハードウェア)に書き込み、処理を終了する。

#### 【0027】

実機のシステムにおいては、ASIC102がASIC103との一連のシリアル通信が完了したタイミングで割込みを発生する(周期は上述の $100\mu s e c \sim 1 m s e c$ )。

#### 【0028】

図5に示すタスク処理では、割込みとは無関係に処理が行われるが、センサの変化が発生しないとモータ停止コマンドは登録されないため、シミュレータ側とフェーズがずれることなく実行が行われる。

#### 【0029】

また、割込み処理においては、シミュレータが割込み要求を行ったタイミングでのみ処理が行われるので、コマンド登録のあふれやレスポンスの2度読み、読み落とし等の問題を起こさずに処理を行うことが可能である。

#### 【0030】

更に問題となるのが、タスク処理におけるタイムアウトである。

#### 【0031】

一般的に制御CPU113はインターバルタイマー等によって時間計測を行っており、機構動作が一定時間で完了しなかった場合にはタイムアウトエラーとしている。ソフトウェアによるシミュレーションは、実機の動作と比較すると低速なので、実機と同じタイムアウト時間では、タイムアウトエラーが発生してしまう。

#### 【0032】

この対処方法としては、制御CPUのタイムアウト値を変更する、基準となるタイマーの周期を長くする、及びシミュレータ側に同期したタイマーに変更するといった方法が考えられる。しかし、前者2つの方法では、シミュレータ実行時間の変動をカバーできず、非常に大きなマージンを見込んだ設定とする必要があるが、この場合には結果としてシミュレーション実行時間が必要以上に長くなる。

ったり、故意にエラーを起こす場合にエラーとなる時間が予測困難といった問題がある。

#### 【0033】

そこで、シミュレータハードウェアまたはパソコン上のソフトウェア（D L L 関数等）でシミュレータからの割込み発生要求をカウントし、一定カウント間隔で模擬タイマー割込みを発生させる。

#### 【0034】

上記した本願発明をまとめると以下の通りである。

#### 【0035】

1、実機の制御システムと同一構成のレジスタ群を制御プログラムを実行するC P U のメモリ空間に実装し、あたかも機構制御A S I C が接続されているかのようにレジスタリード・ライトをシミュレーションする。

#### 【0036】

2、P C ソフトウェアで動作するシミュレータは実機ハードウェアと比較すると定則であるため、制御プログラム側でこれと同期する仕組み（ハードウェアの動きを待たずに勝手にフェーズが進んでしまうのを防ぐ）を持つ。しかも、実機を動作させるプログラムに極力修正を加えずにシミュレータで実行可能なようにする。

#### 【0037】

ここで、上記した実機ハードウェアの一例を図6を参照して説明する。

#### 【0038】

図6に示されるように、主制御部1はC P U 2を備えている。主制御部1は上記したA S I C 1 0 2に相当し、C P U 2は上記した制御C P U 1 0 1に相当する。そのC P U 2には、センサオン／オフメモリ3、レスポンスマモリ5、コマンドメモリ7、ポートON／OFFメモリ6 0が接続されている。これら、センサオン／オフメモリ3、レスポンスマモリ5、コマンドメモリ7、ポートON／OFFメモリ6 0は、上記したレジスタ群に相当する。センサオン／オフメモリ3は、シリアル－パラレル変換器4を介してシリアル回線5 2に接続されている。レスポンスマモリ5は、シリアル－パラレル変換器6を介してシリアル回線5

3に接続されている。コマンドメモリ7は、パラレルシリアル変換器8を介してシリアル回線54に接続されている。

#### 【0039】

また、主制御部1は、アドレス同期信号発生部9を備えている。このアドレス同期信号発生部9は、シリアル回線51に接続されている。

#### 【0040】

ユニット制御部20は選択手段としてスイッチ21を備えており、そのスイッチ21には複数のセンサS<sub>a</sub>, S<sub>b</sub>, … S<sub>n</sub>が接続されている。ユニット制御部20は、上記したASIC103に相当する。さらに、スイッチ21は、センサ切換タイミング生成部40から供給されるタイミング信号に基づいて、時分割のスキャンを繰り返し、各センサの信号（以下、センサ信号と称す）を順次に選択して出力する。

#### 【0041】

上記スイッチ21で選択される各センサ信号のレベルは、A/Dコンバータ22でデジタルデータに変換される。そして、このデジタルデータは、センサレベルデータとしてセンサレベルメモリ23に保持されると共に、比較器24にも供給される。

#### 【0042】

比較器24は、A/Dコンバータ22からの各センサレベルデータとスライスレベルメモリ25に予め保持されている複数のスライスレベルとを比較する。そして、この各比較結果は、比較結果メモリ26に保持される。スライスレベルメモリ25は、センサ切換タイミング生成部40から供給されるタイミング信号に基づき、スイッチ21のスキャンと同じタイミングで、各センサに対応するスライスレベルを順次に出力する。

#### 【0043】

比較結果メモリ26内の各比較結果は、センサスキャンとは独立した図示しないタイミング信号に応じて順次に出力され、パラレルシリアル変換器31でシリアル信号に変換される。こうして変換されたシリアル信号は、上記シリアル回線52を介して、主制御部1のシリアルパラレル変換器4に伝送される。

## 【0044】

センサレベルメモリ23内の各センサレベルデータは、センサスキャンとは独立した図示しないタイミング信号に応じて順次に読出される。尚且つ、後述するコマンド解析部36からの指示に応動するセレクタ32により選択された後、パラレルシリアル変換器33でシリアル信号に変換される。こうして変換されたシリアル信号は、上記シリアル回線53を介して、主制御部1のシリアルパラレル変換器6に伝送されることになる。

## 【0045】

シリアルパラレル変換器34は、主制御部1のパラレルシリアル変換器8からシリアル回線54を介して伝送されるコマンドをパラレル変換する。こうしてパラレル変換されたコマンドは、コマンドメモリ35に保持され、その保持内容がコマンド解析部36によって解析される。

## 【0046】

また、コマンド解析部36は、コマンドメモリ35内の所定のコマンドを解析することにより、センサレベルメモリ23内のセンサレベルデータを主制御部1に伝送させるべく、セレクタ32に指示を与える。

## 【0047】

さらに、コマンド解析部36は、コマンドメモリ35内の所定のコマンドから複数のスライスレベルを解析して、解析結果をスライスレベルメモリ25に保持させる。

## 【0048】

また、コマンド解析部36は、主制御部1から伝送されるコマンドを受信したときに、受信したのと同じコマンドをセレクタ32およびパラレルシリアル変換器33を介して主制御部1に即時に返送（即ち、エコーバックチェック用の返送コマンド）する制御手段を備える。

## 【0049】

同期信号受信部30は、シリアル回線51を介して主制御部1のアドレス同期信号発生部9に接続されており、アドレス同期信号発生部9から供給される同期信号を受信する。

## 【0050】

また、CPU2に接続されたポートON/OFFメモリ60は、パラレルシリアル変換器61を通してシリアル回線62に接続されている。更に、シリアル回線62は、ユニット制御部20内のシリアル-パラレル変換器63を通して、出力ポート回路64に接続されている。この出力ポート64には、ソレノイドPa、DCモータPb、表示器Pnに接続されている。

## 【0051】

アドレス・同期信号発生器9の出力信号はシリアル回線51に接続されるとともに、パラレルシリアル変換器61にも接続されている。同期信号SYNCが低レベルのときにパラレルシリアル変換器61はアドレス・同期信号発生器9から出力されるアドレス信号(A0~A3)をシリアル回線62にSDA信号として出力する。

## 【0052】

一方、ユニット制御部20のシリアル-パラレル変換器63は、シリアル回線62からRDA信号を受信し、アドレス解析部99と出力ポート回路64とに出力する。アドレス解析部99は同期信号受信部30からのSYNC信号に同期してSYNC信号が低レベルのとき、RDA信号のアドレス(A0~A3)が自己のユニット制御部に対するアドレス信号であるか否かを解析する。

## 【0053】

また、出力ポート回路64はアドレス解析部99が自己のアドレスであると解析したとき、同期信号SYNCの高レベルに同期して、RDA信号を出力ポートデータとして取り込むようになっている。

## 【0054】

ユニット制御部20内には、モータ制御回路65も配設されており、該モータ制御回路65には、ステッピングモータMa~Mnが接続されている。

## 【0055】

以下、モータ制御回路65の動作について更に詳細に説明する。

## 【0056】

このモータ制御回路65は、主制御部1側から、シリアル回線54を介して、

モータの初期速度、最高速度、加速レート、減速レート、動作量等のパラメータを与え、動作開始、動作停止等のコマンドを与えることで制御される。

#### 【0057】

そこで、CPU2は、先ずモータ制御回路65に送信したいパラメータやコマンドをコマンドメモリ7に書き込む。パラレルシリアル変換器8は、このコマンドメモリ7に書き込まれた各種パラメータやコマンドを含む情報を読み出し、シリアル信号に変換して、シリアル回線54を介して、シリアル-パラレル変換器34に伝送する。このシリアル信号は、このシリアル-パラレル変換器34でパラレル信号に変換された後、コマンドメモリ35に書き込まれる。その内容はコマンド解析部36がセンサ回路制御コマンド（センサレベルリード、スライスレベル設定コマンド）と同様に解析される。そして、パラメータ及びコマンドがモータ制御回路65に送られるべきものである場合には、当該パラメータ及びコマンドがモータ制御回路65へと送信される。モータ制御回路65では、こうして送られてきたパラメータ及びコマンドに従った動作が行われる。

#### 【0058】

また、上記パラメータ及びコマンドが動作結果の返信を必要とするものである場合には、モータ制御回路65により、その動作結果がセレクタ32へと送信される。上記コマンド解析部36は、同時にセレクタ32を制御して、モータ制御回路65からの動作結果をパラレルシリアル変換器33に送り、当該パラレルシリアル変換器33にてシリアル信号に変換する。このシリアル信号はシリアル回線53を介して主制御部1側のシリアル-パラレル変換器6に送られ、パラレル信号に変換された後、レスポンスマモリ5に保存される。これにより、CPU2は、モータ制御回路65のレスポンスの読み取りが可能となる。

#### 【0059】

次に、出力ポート回路64の動作について詳細に説明する。

#### 【0060】

CPU2は、ポートON/OFFメモリ60のON、又はOFFしたい出力ポートに対応するアドレスに、ONする場合は「1」、OFFする場合は「0」を書き込む。パラレルシリアル変換器61は、ポートON/OFFメモリ60の

内容をシリアル化し、シリアル回線62を通してシリアル-パラレル変換器63に伝送する。こうして、当該シリアル-パラレル変換器63でパラレル化された出力ポートON/OFF情報は、出力ポート回路64によって読み取られる。そして、当該出力ポート回路64は、この出力ポートON/OFF情報に従って、所定のポートの出力を設定することになる。

#### 【0061】

そして、上述したモータ制御回路65の場合と同様に、動作結果の返信が必要な場合は、出力ポート回路64により、その動作結果がセレクタ32へと送信される。上記コマンド解析部36は、同時にセレクタ32を制御して、上記動作結果をパラレル-シリアル変換器33に送り、当該パラレル-シリアル変換器33にてシリアル信号に変換する。このシリアル信号は、シリアル回線53を通して主制御部1側のシリアル-パラレル変換器6に送られ、パラレル信号に変換された後、レスポンスマモリ5に保存される。これにより、CPU2は、出力ポート回路84のレスポンスの読み取りが可能となる。

#### 【0062】

ここで、上記説明した内容を総括すると、本制御システムでは、センサSa～Snに対するコマンド、出力ポート回路64に対するコマンド、モータ制御回路65に対するコマンドのいずれもが、主制御部1から、同一のシリアル回線54を通して、ユニット制御部20側に送信される。

#### 【0063】

即ち、いずれも主制御部1のCPU2の制御の下、各コマンドは、コマンドメモリ7に記憶され、パラレル-シリアル変換器8でシリアル信号に変換された後、シリアル回線54を通して、ユニット制御部20側のシリアル-パラレル変換器34に送信される。そして、当該シリアル-パラレル変換器34にてパラレル信号に変換され、コマンドメモリ35に記憶される。さらに、後段のコマンド解析部36にて、当該制御コマンドが何れの目的に関するものかが判別され、該当する各部に送られることになる。そして、コマンドを受けた各部では、当該コマンドに基づいた所定の動作が行われることになる。

#### 【0064】

また、上記各部に送られたパラメータ及びコマンドが、動作結果の返信を必要とする内容を含むものである場合には、セレクタ32、パラレルシリアル変換器33、シリアル回線53、シリアル-パラレル変換器6を介して、レスポンスマモリ5に各部のレスポンスが保存される。これにより、CPU2は、各部のレスポンスの読み取りが可能となる。

#### 【0065】

このように、同一のシリアル回線54を介して、センサSa～Snの動作状態を制御するためのコマンド、出力ポート回路64の動作状態を制御するためのコマンド、モータ制御回路65に対するモータの回転開始、停止等を制御するためのコマンドを送信することが可能となっている。また、レスポンスについても、同一のシリアル回線53を介して、センサSa～Snからのレスポンス、出力ポート回路64からのレスポンス、モータ制御回路65からのレスポンスを送信することが可能となっている。

#### 【0066】

また、本制御システムでは、センサSa～Snからの出力信号を、スイッチ21を介して順次選択出し、これを受けたA/Dコンバータ22にて当該出力信号をデジタル信号に変換する。そして、比較器24にて、当該デジタル信号とセンサレベルメモリ23に予め記憶されたスレッショルドレベルとを比較する。そして、この比較結果を比較結果メモリ26に記憶した後、パラレルシリアル変換器31にてシリアル信号に変換し、シリアル回線52を介して、主制御部1側のシリアル-パラレル変換器4に送信する。そして、当該シリアル-パラレル変換器4にて、パラレル信号に変換し、センサオン/オフメモリ3に記憶する。

#### 【0067】

##### 【発明の効果】

この発明によれば、実機が完成していないなくてもあたかも実機があるかのように制御プログラムのデバッグが行え、更に様々な異常状態を任意に発生させて制御プログラムを検証することが可能なシミュレータ及びシミュレート方法を提供することができる。

【図面の簡単な説明】

【図1】

実機を用いたデバックを説明するための実機ハードウェアの概略構成を例示する図である。

【図2】

実機を用いないデバッグを説明するためのシミュレータの概略構成を例示する図である。

【図3】

図2に示すシミュレータの動作（レジスタに対するリード・ライト等の動作）を説明するための図である。

【図4】

制御プログラムによる割り込み処理を示す図である。

【図5】

制御プログラムによるタスク処理を示す図である。

【図6】

実機の一例を示す図である。

【符号の説明】

101…制御C P U

102…A S I C (C P U側)

103…A S I C (ユニット側)

104…シリアル回線

105…モータ

106…ソレノイド

107…センサ

111…パソコン

112…シミュレータボード

113…制御C P U

114…シミュレータハードウェア

114 a…レジスタ群

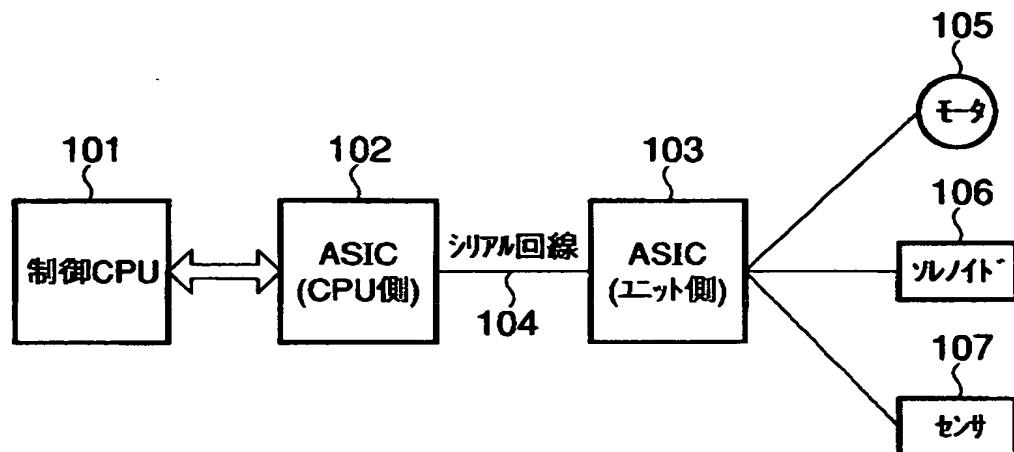
特2000-069232

115…CPU(パソコン側)

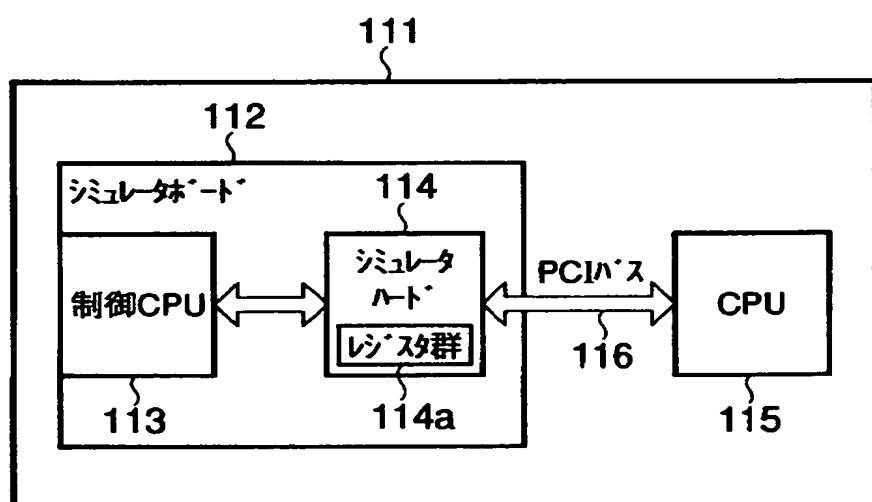
116…PCIバス

【書類名】 図面

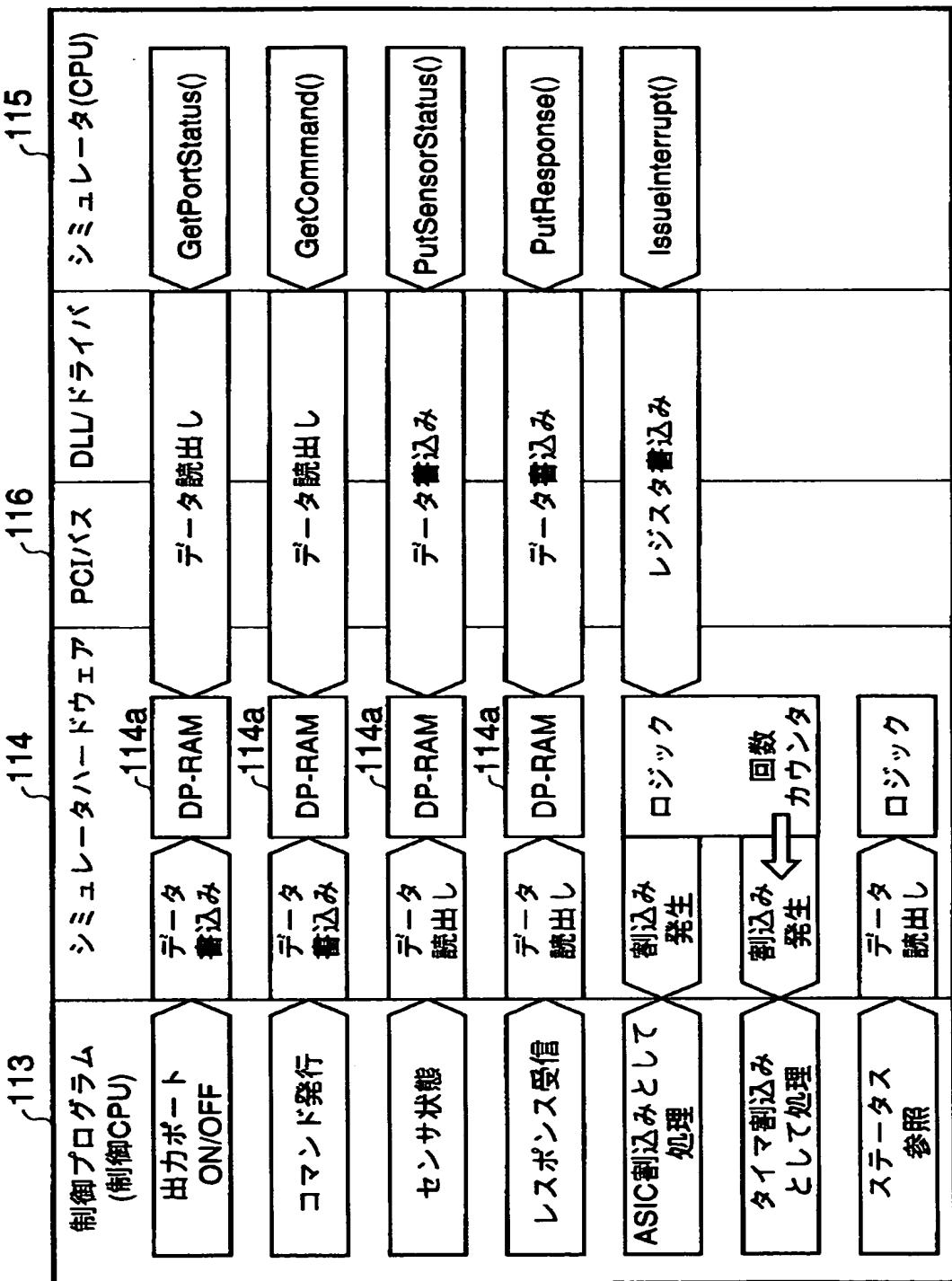
【図1】



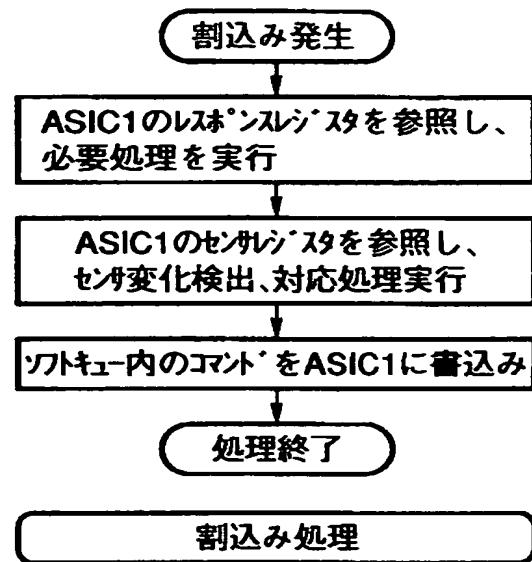
【図2】



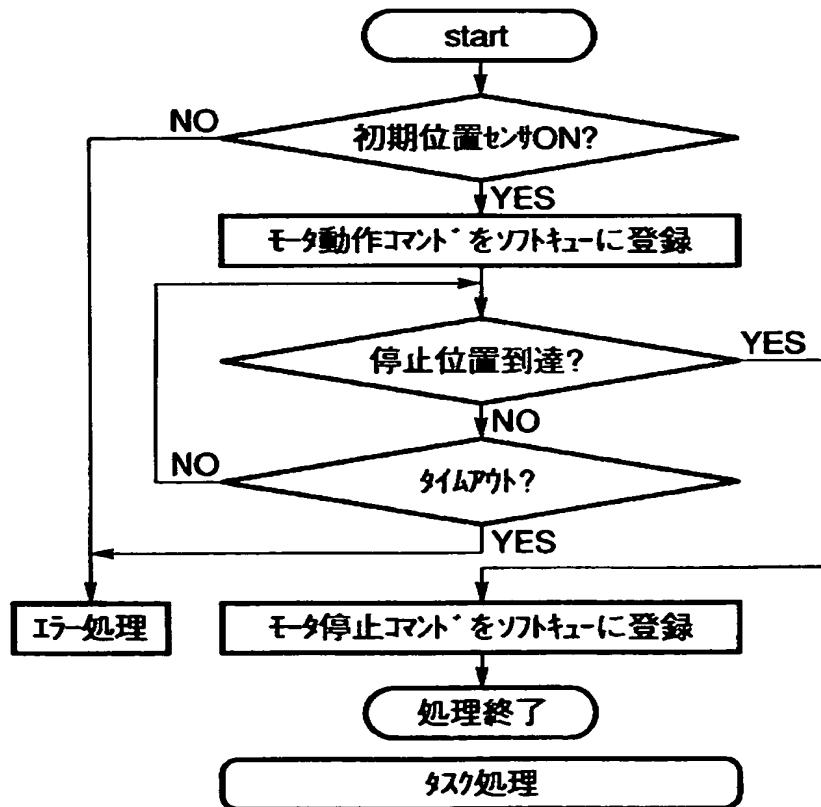
〔図3〕



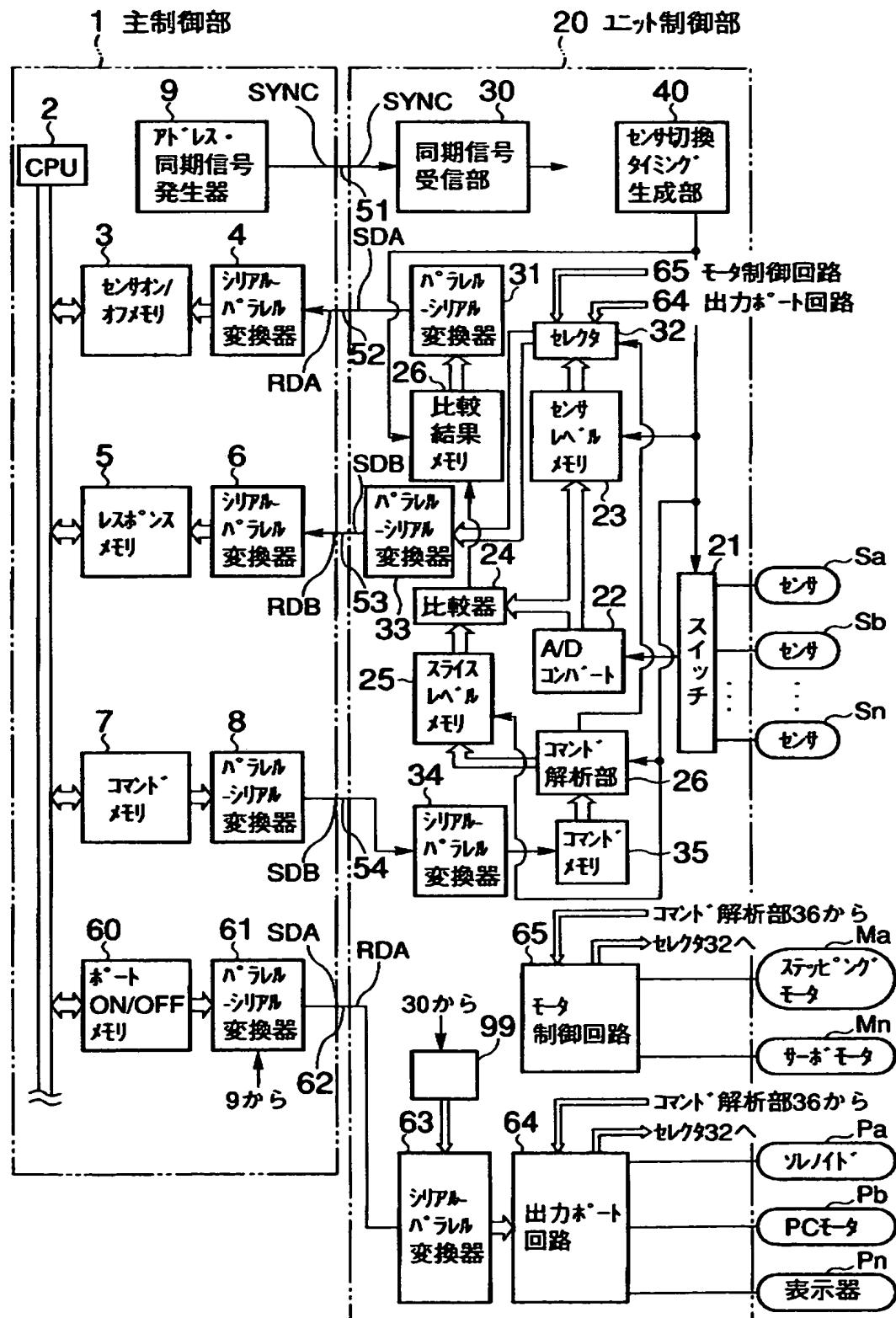
【図4】



【図5】



【図6】



【書類名】 要約書

【要約】

【課題】 実機が完成していなくてもあたかも実機があるかのように制御プログラムのデバッグを行うことが可能なシミュレータを提供すること。

【解決手段】 制御CPU(113)と、シミュレーション用CPU(115)と、制御CPU及びシミュレーション用CPUの両CPUから読書可能なメモリ(114a)と、シミュレーション用CPUとメモリとを接続するバスと(116)を備えている。制御CPUにより制御情報をメモリに書き込み、シミュレーション用CPUによりメモリに書き込まれた制御情報をバスを介して読み出し、この制御情報に基づくシミュレーションを実行し、このシミュレーションの結果をバスを介してメモリに書き込み、制御CPUによりメモリに書き込まれたシミュレーションの結果を読み出す。

【選択図】 図2

出願人履歴情報

識別番号 [000003078]

1. 変更年月日 1990年 8月22日

[変更理由] 新規登録

住 所 神奈川県川崎市幸区堀川町72番地

氏 名 株式会社東芝